TDB-ACC-NO: NN880830

DISCLOSURE TITLE: Conversion of Structured Documents for Filing

PUBLICATION-DATA: IBM Technical Disclosure Bulletin, August 1988, US

VOLUME NUMBER: 31

ISSUE NUMBER: 3

PAGE NUMBER: 30 - 31

PUBLICATION-DATE: August 1, 1988 (19880801)

CROSS REFERENCE: 0018-8689-31-3-30

DISCLOSURE TEXT:

A technique is proposed for converting structured documents, described in a generalized markup language, to a format suitable for filing, searching, and retrieval under a database-oriented document filing system. For document filing under a database-oriented filing system, it is advantageous to consider each document as a hierarchically-structured object, and represent the document by a tree with its nodes representing identifiable components of the document. For each type of document, a schema tree is used to describe the logical structure of the documents, with a document instance of a given type containing hierarchically-related components defined by the respective schema.

With the markup approach to document representation, a document conforming to a specific style is represented by a file containing the document content as well as various imbedded nested tags, each tag identifies a component of the document, provides certain semantics, and specifies the way to format the component. The document can be considered as a hierarchically-structured object with each node of the tree representing a component. Part of the tag definition is to specify which tags can be included in the scope of the subject tag. Considering the tags as nodes and the allowable inclusions as arcs, a directed graph can be drawn to show the acceptable logical forms

info complet

for
documents conforming to this style.
This graph, though containing a
root node, often includes recursions and is in general not a tree.
As a result, the maximum depth of possible document instance trees is
not bounded. Hence, this document model does not map readily to that
of the filing system described above. For the markup model, it is
observed that (1) many high-level tags identify with the same logical
components of a document that are useful for searching/retrieval, (2)
some low-level tags are only useful for formatting and have no value
for searching/retrieval, (3) for searching or for component
retrieval, the usefulness of the identity of a document component
decreases quickly as we proceed down the tree from its root, and (4)
in practice, if a certain (finite) set of components are specified
for a document type for the filing system, it is sufficient to
support most component searching/retrieval functions.
At the time a
document style is defined, a corresponding schema and a mapping from
the tags generated by the style to the nodes of the schema can be
defined by progressively creating an expansion tree by tracing the
arcs of the style graph starting from its root. Each node in this
expansion tree corresponds to a tag in an imaginary, and possibly
infinite, markup document containing all possible components allowed
by the style. Because of recursion in the style graph, multiple
nodes may associate with tags of the same name. But they represent
different document components. If the identity of a tag is of use for
either searching or direct component retrieval, then a corresponding
node is added to the expansion tree. A mapping from the tag to the
new node is recorded.
If a tag carries no useful semantics for
searching/retrieval, the tag is ignored and its scope is considered a
part of its parent node's scope. If all tags imbedded within a tag

scope are deemed not useful for searching/retrieval, the entire scope of this tag and the imbedded tags is considered a single component represented by a leaf node in the ex- pansion tree. Tracing is stopped for the subgraph emanating from this tag. If a tag is useful for capturing individual search keys (e. g., highlighting tag and indexing tag), it is mapped to either an existing node or to a new node in the expansion tree at an appropriate place as desired. The new node represents a new component of the document for storing the captured keys.

When all arcs of the style graph are exhausted, a leaf node is added to each internal node of the expansion tree to hold data not belonging to any lower-level tag. The expansion tree is now complete and becomes the schema tree used by the filing system. The mapping from the relevant tags to the corresponding nodes in the schema tree is saved for use in document instance conversion. During document instance conversion, the markup document is scanned to capture the tags using a parser suitable for the particular markup language. The mapping obtained in the above-described phase is used to identify the relevant tags and convert them to the corresponding components described by the schema tree. Although some tags are not represented as distinct nodes in the schema tree, all tags found in a document instance are stored as data to preserve the original information.

Secrets Act, 18 U.S.C. 1905.